

Analisa hyperparameter dalam model MLP Regressor yang telah dilakukan pada codingan dapat dijelaskan sebagai berikut:

1. **Hidden Layers:**

- Jumlah hidden layer mempengaruhi kemampuan model dalam mempelajari pola yang kompleks dalam data.
- Semakin banyak hidden layer, model dapat mempelajari fitur yang lebih kompleks, tetapi terlalu banyak hidden layer juga dapat menyebabkan overfitting.
- Dalam kode ini, tiga opsi hidden layer diuji: [4], [8, 4], dan [16, 8, 4]. Hasil terbaik akan dipengaruhi oleh kompleksitas dataset dan kebutuhan model.

2. **Activation Function:**

- Fungsi aktivasi menentukan sifat non-linear dari model, yang mempengaruhi kemampuannya dalam mempelajari pola yang kompleks.
- Opsi fungsi aktivasi yang diuji dalam kode ini adalah ReLU, Sigmoid, Tanh, Softmax, dan Linear.
- Pemilihan fungsi aktivasi yang tepat akan bergantung pada sifat data dan tujuan model. Misalnya, Sigmoid dan Tanh cocok untuk masalah klasifikasi, sedangkan ReLU dan Linear cocok untuk regresi.

3. **Learning Rate:**

- Learning rate menentukan seberapa cepat model dapat menyesuaikan bobot parameter selama proses pembelajaran.
- Learning rate yang terlalu tinggi dapat menyebabkan model divergen, sedangkan learning rate terlalu rendah dapat memperlambat konvergensi.
- Dalam kode ini, beberapa opsi learning rate diuji: 10, 1, 0.1, 0.01, 0.001, dan 0.0001. Nilai optimal akan bergantung pada dataset dan arsitektur model.

4. **Batch Size:**

- Batch size menentukan jumlah sampel yang digunakan dalam setiap iterasi pembaruan parameter.
- Batch size yang lebih besar dapat meningkatkan efisiensi komputasi, tetapi batch size yang terlalu besar juga dapat menyebabkan overfitting.
- Dalam kode ini, beberapa opsi batch size diuji: 16, 32, 64, 128, 256, dan 512. Nilai optimal akan bergantung pada dataset dan sumber daya komputasi yang tersedia.

5. **Epochs:**

- Epochs menentukan jumlah kali model dilatih pada seluruh dataset.

- Semakin banyak epochs, model dapat belajar lebih baik, tetapi terlalu banyak epochs juga dapat menyebabkan overfitting.
- Dalam kode ini, beberapa opsi epochs diuji: 1, 10, 25, 50, 100, dan 250. Nilai optimal akan bergantung pada kompleksitas dataset dan konvergensi model.

Dalam proses pemodelan, penting untuk melakukan uji coba berbagai kombinasi hyperparameter dan menganalisis hasilnya untuk menemukan konfigurasi terbaik yang sesuai dengan dataset dan tujuan pemodelan. Kode di atas menggunakan grid search untuk mengevaluasi berbagai kombinasi hyperparameter dan menyimpan hasilnya dalam sebuah DataFrame untuk analisis lebih lanjut.

Dampak masing-masing hyperparameter terhadap output MLP Regressor

#### 1. **Hidden Layers:**

- **Dampak:** Jumlah dan konfigurasi hidden layers berpengaruh langsung terhadap kemampuan model dalam menangkap pola kompleks. Penambahan lapisan dapat meningkatkan kapasitas model, tetapi jika terlalu banyak, risiko overfitting meningkat, terutama jika data pelatihan tidak cukup. Model mungkin belajar noise dalam data daripada pola yang sebenarnya, sehingga performa pada data baru bisa menurun.

#### 2. **Activation Function:**

- **Dampak:** Fungsi aktivasi menentukan seberapa baik model dapat belajar representasi non-linear dari data. ReLU, misalnya, membantu menghindari masalah vanishing gradient, yang umum terjadi pada fungsi sigmoid dan tanh. Jika fungsi aktivasi tidak cocok dengan sifat data, model mungkin tidak dapat belajar dengan baik, yang berujung pada output yang buruk.

#### 3. **Learning Rate:**

- **Dampak:** Learning rate yang terlalu tinggi bisa menyebabkan model meleset dari minimum global, sedangkan yang terlalu rendah dapat memperlambat proses konvergensi. Keduanya dapat menghasilkan output yang tidak optimal. Learning rate yang tepat membantu model mencapai konvergensi dengan efisien dan menghasilkan prediksi yang akurat.

#### 4. **Batch Size:**

- **Dampak:** Ukuran batch mempengaruhi stabilitas dan kecepatan pelatihan. Ukuran batch kecil memberikan pembaruan parameter yang lebih sering, yang bisa membantu model belajar lebih baik, tetapi bisa membuat pelatihan lebih lambat. Sebaliknya, ukuran batch besar dapat mempercepat pelatihan tetapi berisiko menyebabkan model terjebak dalam minimum lokal, yang dapat memperburuk output.

## 5. Epochs:

- **Dampak:** Jumlah epoch menentukan seberapa sering model melihat data pelatihan. Terlalu sedikit epoch bisa membuat model tidak belajar dengan baik, sementara terlalu banyak dapat menyebabkan overfitting, di mana model belajar terlalu baik pada data pelatihan dan gagal generalisasi pada data baru. Penentuan jumlah epoch yang tepat penting untuk mencapai keseimbangan antara belajar dan generalisasi.